REMARKS

Pending Claims:

In this application, claims 1-62 are currently pending. Claims 2-3, 5-15, 17-35, 37-45, and 47-55 have not be altered since filing. Claims 1, 4, 16, 36, and 46 are amended by this Response. Claims 56-62 have been added. Entry of these amendments is respectfully requested.

Claim Objections

The Examiner objected to claims 4 and 16 for being in improper dependent form. The applicant has amended these claims to correct his problem. The applicant has also amended claim 36 to correct a typographical error.

Rejection under 35 U.S.C. §103

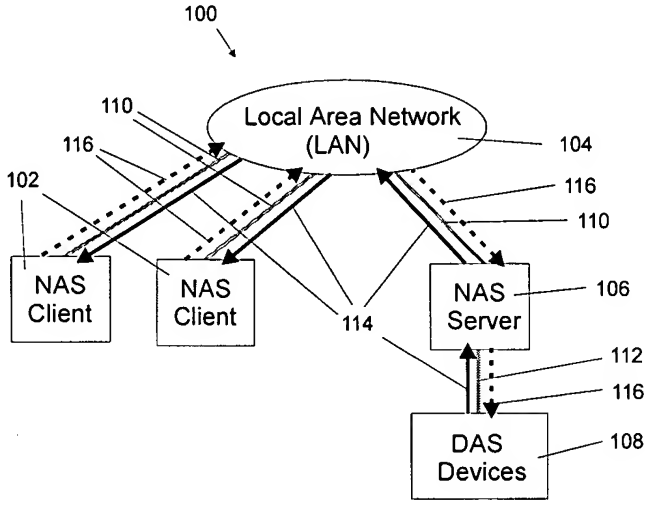
The Examiner has rejected most of the prior claims as being unpatentable over Row (U.S. Patent No. 5,802,366) and Philbrick (U.S. Patent Application Publication No. US2001/0037406), with the remainder of the prior claims being rejected as obvious over Row and Philbrick further in light of Gregorson (U.S. Patent No. 5,758,342) and/or Henson (U.S. Patent No. 5,202,971). The Applicant respectfully traverses these rejections. To explain the difference between the present claims and the cited prior art, this response will first discuss the admitted prior art described in the specification, explain the present invention, and then discuss the prior art cited in this office action in light of the pending claims.

Acknowledged Prior Art Described in the Specification.

The best way to understand the present invention is to understand the prior art file systems explained in the specification. As described in the Specification in paragraphs [0021] to [0050] and Figures 1 and 2, prior art file systems can be divided between local files systems and distributed file systems. Distributed file systems can be divided between network attached storage (NAS-based) distributed file systems where storage devices are locally attached to servers, and storage area network (SAN-based) distributed file systems where storage devices are shared between client computers.

Figure 1 of the Specification shows the data-paths and components of a typical, prior art NAS-based file sharing environment, and is repeated below for the convenience of the Examiner.

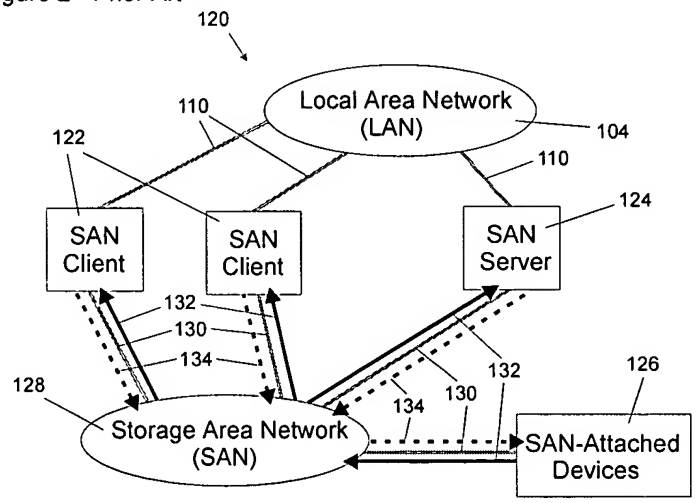
Figure 1 - Prior Art



In this environment 100, clients 102 connect to a server 106 via a local area network 104 (using network-based I/O interface links 110). The server 106 connects to direct attached storage device 108 via channel-based I/O interface links 112. Both the read data-path (solid line 114) and the write data-path (dotted line 116) pass between the clients 102 and the storage device 108 through both the LAN 104 and the server 106.

Figure 2 of the Specification shows the data-paths and components of SAN-based distributed file sharing environment that utilizes a server:

Figure 2 - Prior Art



While clients 122 in this environment 120 are connected to the server 124 via LAN 104, only control and consistency information passes across the LAN 104 to server 124. Some SAN-based distributed file systems do not utilize server 124. When a server is used, the server performs all namespace and metadata operations. In a serverless SAN-based file system, the SAN server 124 and the LAN 104 are unnecessary and the clients 122 perform local file management tasks. Whether or not a server 124 is used, both the read data-path (solid line 132) and the write data-path (dotted line 134) in the SAN-based distributed file sharing environment 120 pass between the clients 122 and the storage devices 126 over the SAN 128 without passing through the server 124.

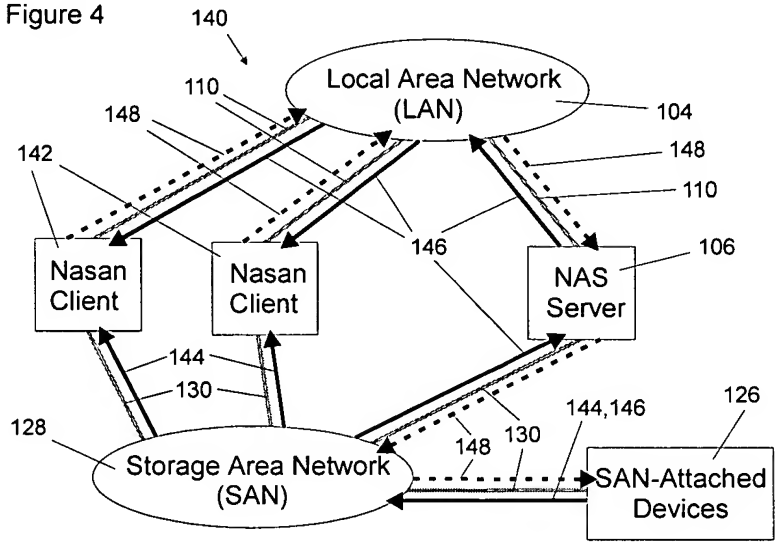
Summary of the Present Invention

The present invention file system is a distributed file system that differs from all prior art file systems in that some file requests are handled by a server, while other file requests pass directly over the storage area network without passing through the server. As stated in the summary of the invention:

> **[0063]** The present invention is a distributed file system that utilizes aspects of a NAS server system along with a storage area network having at least one SAN-attached storage device. By combining these two architectures, it is possible to achieve the benefits of fast data reads over a SAN as well as some of the consistency benefits of using a NAS server. The present invention combines these two architectures by creating separate data paths for write and read requests.

This system is shown in Figure 4:

Figure 4



The summary of the invention continues:

> **[0064]** The write data-path of the present invention *[DOTTED PATH 148]* is similar to the write data-path of prior art NAS, with the DAS storage device being replace with a SAN-attached storage device accessed over a SAN. This is accomplished so that all write activities to the SAN attached storage device are serialized through one server, while still allowing each client write access to the volume stored on the SAN-attached device.
>
> **[0065]** The primary read data-path of the present invention *[SOLID PATH 144]* is similar to the read data-path of prior art SAN environments, whereas the secondary read data-path *[SOLID PATH 146]* is similar to the read data-path of prior art NAS environments. Since most reads pass directly from the SAN-attached storage device to the clients, the present invention takes full advantage of high-speed SAN protocols. In those rare instances where the primary read data path is not available, the present invention can utilize the secondary data path of typical NAS environments.

[Bracketed information inserted for the benefit of the Examiner].

As shown in Figure 4 and explained in the Specification, the present invention is unique in that write requests from an application are submitted from the client to a server where the server stores the data on the storage device 126, while read requests are handled locally at the clients by retrieving the data directly across the SAN 128. To accomplish this, the server requests and analyzes the metadata stored on the storage device during write operations, while the clients request and analyze this metadata locally during read operations. No other prior art file system divides write and read requests in this manner.

Claim 1.

Claim 1 claims a files system for handling read and write requests to a SAN-attached storage device having the following elements: a NAS server; a local component, a remote component, and an upper level component. The upper level component communicates with application programs and is responsible for dividing read and write requests. All write requests are submitted to the remote component, and at least some read requests are submitted to the local component. The local component receives these read requests, and communicates with the SAN-attached storage device over the storage network. To clarify how the local component handles the read requests, claim 1 has been amended to state that the local component interprets metadata stored on the storage device. The remote component communicates with the NAS server for handling the write requests.

Row is cited by the Examiner as disclosing a remote component that communicates with the NAS server over a local area network and an upper level component that submits all write requests to the remote component and at least some

read requests to the local component. The Examiner acknowledges that Row fails to disclose a local component and a NAS server that communicates with the SAN.

The applicant respectfully points out that Row describes a "new, server-specific I/O architecture that is optimized for a Unix file server's most common actions—file operations." Row, col. 4, lines 16-18. As such, Row describes an improved file server such as the NAS Server 106 shown above in Figure 1 of the present invention. There is no remote component that communicates with another (NAS) server in Row. The cited section of Row for teaching this element (col. 12, lines 39-49) describes a microprocessor data bus 212 and related FIFOs 240, 260, and 270. No server is mentioned in this section, nor is any communication with that server over a local area network.

As for the upper level component, the Applicant respectfully submits that since Row does not teach a local component (as acknowledged by the Examiner), Row cannot teach an upper level component that submits at least some read requests to the local component. As explained above, the present invention improves the prior art by submitting write requests to a server (via the remote component) and by handling some read requests locally (via the local component). The upper-level component that divides the file requests between the remote component and local component is an important part of this invention. Since Row does not teach an upper level component that divides file requests in this matter, Row cannot be considered to teach or suggest the upper level component of claim 1. The cited sections of Row (col. 6, line 58- col. 7, line 11; col. 10, lines 17-32, and col. 20, lines 36-44) do not suggest otherwise. The text from column 6 to column 7 describes the ability of the file server of Row to handled NFS procedure calls. The text on column 10 describes how a network controller 110 receives a read NFS request from a client and then submits a LNFS (local NFS) request to the file controller 112. Similarly, the text in column 20 describes how a write command received by a network controller is handled by the file controller 112. In effect, the cited sections of Row describe how Row is acting like a traditional NAS server 106 of the present application Figure 1, where the central server communicates with a client over NFS and stores data locally using a local file system. This type of prior art is described in the specification of the present application at paragraph [0030].

Philbrick is cited for teaching i) a local component that communicates with the SAN-attached storage device over a SAN and ii) a NAS server that communicates with the SAN-attached storage device over the SAN. However, the cited sections of Philbrick

teach only the existence of a SCSI controller on a PCI bus to communicate with a storage device (paragraph [0044]), a Fibre Channel controller on a PCI bus that communicates with a Fibre Channel SAN (paragraph [0009]) and a traditional NAS server (paragraphs [0004]-[0009]).

Thus, Row and Philbrick do not teach the four elements of claim 1. In fact, neither reference seems as directly relevant to claim 1 as the admitted prior art shown in Figures 1 and 2 of the specification of the current application. The applicant acknowledges that Figure 1 contains "a NAS server" and that the clients 102 contain "a remote component that communicates with the NAS server over a local area network." Furthermore, Figure 2 shows a client 122 that contain "a local component that communicates with the SAN-attached storage device over a storage area network." What no prior art shows or suggests is an upper level component that divides requests between the remote component and the local component. In addition, no prior art shows the simultaneous use of a server for handling write requests and a local component that handles read requests and is capable of interpreting file system metadata. Thus claim 1 must be considered patentable over the prior art.

The Other Independent Claims.

Claims 17 and 46 each claim a client computer with either a remote "file system" or a remote "component" that communicates with and makes file requests to a server computer. The client computer also contains a local "file system" or "component" that communicates with a SAN-attached device over a SAN. In addition, both claims include an upper level "file system" or "component" that services file requests from an application program operating on the client computer and that divides the requests between the remote file system/component and the local file system/component. As explained above, this division of file system requests between the local and remote file systems/components operating on a client computer is unique to the present invention.

The method claim 34 requires the reception of a file request from an application, and a determination as to whether the request is a local request or a remote request. Local requests are handled through direct access to the SAN-attached device over a SAN. Remote requests are handled by a server computer accessible by a LAN. Once

again, this division of file requests between local requests and remote requests and the separate way in which these requests are handled are unique to the present invention.

New Claims 56-62.

Newly added claim 56-59 are directed toward a system that has client software on a client computer, server software on a server computer, where the server writes real-data on a SAN-attached storage device for the client, and where the client receives real-data relating to a read request directly from the SAN-attached storage device. These claims are supported by the specification as filed, and are not taught in or suggested by the prior art.

Claims 60 and 61 are new dependent claims depending on claim 1. Claim 62 is a new dependent claim depending on claim 17. These claims are also supported by the specification as filed.

CONCLUSION

All of the claims remaining in this application should now be seen to be in condition for allowance. The prompt issuance of a notice to that effect is solicited.

Respectfully submitted,
DATAPLOW, INC.
By its attorneys:

Date: February 17, 2006

Daniel A. Tysver
Registration No. 35,726
Beck & Tysver, P.L.L.C.
2900 Thomas Ave., #100
Minneapolis, MN 55416
Telephone: (612) 915-9634
Fax: (612) 915-9637